

REMARKS

Claims 1, 10 and 43-53 have been amended. Claims 1 - 53 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Drawing Objections:

The Examiner objected to the drawings for failing to comply with 37 CFR 1.84(p)(5) because the specification failed to include a description of Fig. 21. The specification has been amended to include a short description of Fig. 21. No new subject matter has been added. Thus, applicants submit that the drawings fully comply with 37 CFR 1.84(p)(5).

Section 101 Rejection:

The Examiner rejected claims 1-21 and 43-53 under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Applicants traverse this rejection. However, to expedite prosecution of the application, independent claims 1 and 10 have been amended to recite computer-implemented methods and claims 43-53 have been amended to recite a tangible, computer accessible medium. Applicants respectfully request removal of the § 101 rejections.

Section 103(a) Rejections:

The Examiner rejected claims 1, 2, 6-9, 22, 23, 27, 30, 31 and 43-46 under 35 U.S.C. § 103(a) as being unpatentable over Tuatini (U.S. Publication 2002/0035645) in view of East, et al. (U.S. Patent 5,321,841) (hereinafter “East”). Applicants respectfully traverse this rejection for at least the reasons presented below.

The rejection is improper because the Examiner has not shown that Tuatini qualifies as a prior art reference. The Examiner has the burden of proof to produce the

factual basis for the rejection. *In re Warner*, 154 USPQ 173, 177 (C.C.P.A. 1967), *cert. denied*, 389 U.S. 1057 (1968). Since the Examiner has not proven that Tuatini qualifies as a prior art reference, the Examiner has not met this burden of proof and the rejection is improper. More specifically, Tuatini's published application was filed on December 28, 2000, after Applicants' filing date of September 15, 2000. Tuatini does claim the benefit of a provisional application filed December 30, 1999. However, the December 30, 1999 filing date can only be used as Tuatini's prior art date for the subject matter that is common to both Tuatini's published application and the provisional application. Since it is common practice for a later filed utility application to include more or different subject matter than its earlier provisional application, it is unclear whether the material in Tuatini relied upon by the Examiner was actually present in Tuatini's provisional application. The Examiner must show that the subject matter on which the Examiner is relying on to reject Applicants' claims is also present in Tuatini's provisional application. Until the Examiner has made this showing, the rejection is improper. *See, In re Wertheim*, 209 USPQ 554 (CCPA 1981).

Moreover, Tuatini's published application is not entitled to the December 30, 1999 date as a prior art date unless at least one claim of Tuatini's published application is supported (under 35 U.S.C. § 112) in the provisional application. Under 35 U.S.C. 119(e)(1), a published utility application is not entitled to its provisional application's filing date as a prior art date unless at least one claim of the published utility application is supported (per 35 U.S.C. § 112) in the provisional application. The rejection is improper unless the Examiner can show that Tuatini's published application has the necessary claim support in the provisional application to be entitled to the provisional application's filing date as its prior art date. *See also* M.P.E.P. § 2136.03(IV).

Since the Examiner has not shown that both of the above requirements are met for Tuatini to qualify as prior art to the present application, the current rejection is improper.

Even if Tuatini does qualify as prior art, Tuatini in view of East fails to teach or suggest generating a computer programming language object from a data representation language representation of the object, as recited in claim 1. The Examiner cites paragraphs [0079 –0082] of Tuatini. However, the cited passage does not describe generating a computer programming language object from a data representation language *representation of the object*. In contrast, Tuatini describes a serialization service for converting XML data into a Java object and vice versa. The XML data and data types described in the cited passage do not refer to data representation language representations of objects. Instead, Tuatini teaches that the serialization service is used to serialize and deserialize request messages (Tuatini, paragraph [0079]). Tuatini is clearly teaching that XML data, such as in a request message, may be converted into Java objects. Tuatini goes on to describe using the serialization service for deserializing an XML formatted document 3202, messages, and configuration files. No mention is made of any data representation language representations of computer programming objects, nor of generating a computer programming language object from a data representation language representation of the object.

Additionally, Tuatini in view of East fails to teach or suggest deleting the computer programming language object in response to the terminating access. The Examiner admits that Tuatini does not teach deleting the computer programming language object in response to the terminating access, and relies upon East, citing column 15, lines 60-68. However, this passage of East refers to a Shutdown field 428 of an object type descriptor (OTD). East teaches that the Shutdown field 428 points to a shutdown procedure that is called once when an object type is permanently removed from the system, generally at shutdown time. The cited passage is discusses a specific shutdown procedure called when an object *type* is removed from a system. Removing an object type is not the same as deleting a computer programming language object itself. Additionally, East teaches that his shutdown procedure is called in response to a device shutdown or otherwise when the object type is permanently removed from a system. No mention is made of a user terminating access to a client device, or of deleting a computer programming language object in response to the user terminating access. Merely

specifying a shutdown procedure to be called when an object type is permanently removed or when a machine is shutdown does not teach anything about deleting a computer programming language object itself in response to a user terminating accessing a client device. As Tuatini also fails to teach deleting the computer programming language object in response to terminating access, the Examiner's suggested combination of Tuatini and East fails to teach or suggest deleting the computer programming language object in response to terminating access.

Thus, for at least the reasons above, the rejection of claim 1 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks to those above regarding claim 1 also apply to claims 22 and 43.

Regarding claim 2, Tuatini in view of East fails to teach or suggest the client device receiving a message in the data representation language from a service device in the distributed computing environment prior to said generating a computer programming language object. The Examiner cites paragraphs [0079] and [0083] of Tuatini. However, the cited passage refers to a service device receiving a message from a client device. Tuatini is describing creating and returning a Java object representing a client request messages. Tuatini specifically refers to converting the received message from a client format (paragraph 0083). Thus, the cited passage fails to teach or suggest a client device receiving a message in a data representation language *from a service device*.

Tuatini in view of East also fails to teach or suggest wherein the message includes the data representation language representation of the object. The Examiner cites paragraph [0083] of Tuatini. However, this cited passage specifically refers to deserializing a message and generating “the Java object *representing the message*” (italics added). As described above regarding claim 1, Tuatini teaches converting XML data to Java objects. The cited passage describes generating a Java object that represents a received message. Tuatini does not teach or suggest that the received message includes a data representation language representation of a computer programming language object. Instead, Tuatini teaches the use of Java objects generated from XML data to

allow the XML data, such as the received message, to be accessed manipulated as a Java object. This is quite different from a message including a data representation language representation of a computer programming language object.

East does not teach or suggest anything regarding a client device receiving a message in the data representation language from a service device in the distributed computing environment prior to generating a computer programming language object, wherein the message includes the data representation language representation of the object, and thus fails to overcome any of the above mentioned deficiencies of Tuatini.

Therefore, the rejection of claim 2 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks apply to claims 23 and 44.

In regard to claim 7, Tuatini in view of East does not teach or suggest generating a plurality of computer programming language objects from data representation language representations of the objects. The Examiner cites paragraphs [0079-0082] of Tuatini. However, as described above, the cited passage fails to teach generating computer programming language objects from data representations of the objects. Instead, as described above regarding claim 1, Tuatini only teaches converting XML data into Java objects, but does not teach generating computer programming language objects from data representation language *representations* of the objects. Converting XML data into Java objects is quite different from generating a computer programming language object from a representation of the object. For a more detailed discussion regarding Tuatini's failure to teach generating a computer programming language object from a data representation language representation of the object, please refer to the discussion of claim 1 above.

Tuatini in view of East further fails to teach or suggest deleting the plurality of computer programming language objects in response to said terminating access. The Examiner cites column 15, lines 60-68 of East. However, as noted above regarding claim 1, this passage of East describes a shutdown field in an object type descriptor that points to a shutdown procedure called when an object type is permanently removed from a

system, such as at system shutdown. The cited passage does not mention deleting any computer programming language objects in response to a user terminating accessing a client device. For a more detailed discussion, please refer to the remarks above regarding claim 1.

The rejection of claim 7 is not supported by the prior art and removal there of is respectfully requested. Similar remarks apply to claims 27 and 45.

The Examiner rejected claims 3-5, 10-21, 24-26, 28, 29, 32-42 and 47-52 under 35 U.S.C. § 103(a) as being unpatentable over Tuatini and East in view of Wu (U.S. Patent 5,774,551). Applicants respectfully traverse this rejection in light of the following remarks.

As with the rejection of claims 1, 2, 6-9, 22, 23, 27, 30, 31 and 43-46, discussed above, this rejection is improper until the Examiner proves that Tuatini qualifies as a prior art reference. Please see Applicants' arguments above regarding Tuatini's provisional application priority date.

Furthermore, regarding claim 10, Tuatini in view of East, in further view of Wu fails to teach or suggest the client device receiving a message in a data representation language from a service device in the distributed computing environment, wherein the message includes a data representation language representation of a computer programming language object. The Examiner cites paragraphs [0079] and [0083] of Tuatini. However, the cited passage refers to a service device receiving a message from a client device. Tuatini is describing creating and returning a Java object representing a client request messages. Tuatini specifically refers to converting the received message from a client format (paragraph 0083). Thus, the cited passage fails to teach or suggest a client device receiving a message in a data representation language *from a service device*.

Additionally, the cited passage does not mention wherein the message includes a data representation language representation of a computer programming language object. The Examiner cites paragraph [0083] of Tuatini. However, this cited passage specifically refers to deserializing a message and generating “the Java object *representing the message*” (italics added). As described above regarding claim 1, Tuatini teaches converting XML data to Java objects. The cited passage describes generating a Java object that represents a received message. Tuatini does not teach or suggest that the received message includes a data representation language representation of a computer programming language object. Instead, Tuatini teaches the use of Java objects generated from XML data to allow the XML data, such as the received message, to be accessed manipulated as a Java object. This is quite different from a message including a data representation language representation of a computer programming language object.

Neither East nor Wu overcomes the deficiencies of Tuatini discussed above. Thus, the Examiner proposed combination of Tuatini, East and Wu fails to teach or suggest a client device receiving a message in a data representation language from a service device in the distributed computing environment, wherein the message includes a data representation language representation of a computer programming language object. Therefore, the rejection of claim 10 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 10 also apply to claims 32 and 47.

Regarding claim 11, Tuatini in view of East in further view of Wu fails to teach or suggest wherein the message includes access information for the computer programming language object, wherein said determining if the user has access rights to the computer programming language object uses the access information. The Examiner cites column 22, line 65 through column 23, line 5 of East. However, the cited passage describes how access control is checked by comparing the access rights of a user against access control information in an object. No mention is made of a *message* including access information for a computer programming language object or that determining if a user has access rights for the object uses the access information *from the message*. Instead, the cited

passage only refers to access control information in an object itself. These are two different concepts. Tuatini and Wu both fail to overcome Wu's deficiencies regarding a message including accessing information and regarding wherein determining if a user has access rights to the computer programming language object uses the access information. Thus, the rejection of claim 11 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks also apply to claims 33 and 48.

Regarding claim 12, Tuatini in view of East in further view of Wu fails to teach or suggest deleting the computer programming language object in response to said terminating access. The Examiner cites column 15, lines 60 –68 of East. However, as described above regarding claim 1, Tuatini and East, both singly and in combination, fail to teach deleting the computer programming language object in response to the user terminating access. As noted above regarding claim 1, the cited passage refers to a Shutdown field 428 of an object type descriptor (OTD). East teaches that the Shutdown field 428 points to a shutdown procedure that is called once when an object type is permanently removed from the system, generally at shutdown time. The cited passage is discusses a specific shutdown procedure called when an object *type* is removed from a system. Removing an object type is not the same as deleting a computer programming language object itself. Additionally, East teaches that his shutdown procedure is called in response to a device shutdown or otherwise when the object type is permanently removed from a system. No mention is made of a user terminating access to a client device, or of deleting a computer programming language object in response to the user terminating access. Please see the discussion of claim 1 above for a more details regarding Tuatini's and East's failure to teach deleting a computer programming language object in response to the a terminating access.

Wu also fail to teach or suggest anything regarding deleting a computer programming language object in response to the terminating access and thus fails to overcome the deficiencies of Tuatini and Wu described above. Thus, the rejection of claim 12 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks apply to claims 34 and 49.

Regarding claim 16, Tuatini in view of East in further view of Wu fails to teach or suggest the user terminating said accessing the client device; and storing the computer programming language object in response to said terminating access. The Examiner cites paragraph [0051] of Tuatini and refers to how Tuatini's action handler stores a response message in an action response object after the action handler completes performance of its business logic. However, as noted by the Examiner, Tuatini's action handler stores a response message in an action response object after performing business logic. Storing a response message in an action response object after completion of business logic processing is not the same as, nor does it suggest, storing a computer programming language object in response to a user terminating accessing a client device. Tuatini make no mention of storing any objects in response to a user terminating accessing a client device. Additionally, neither East nor Wu, teaches or suggests anything about storing the computer programming language object in response to a user terminating access. Thus, the combination of Tuatini, East and Wu also fails to teach storing the computer programming language object in response to a user terminating access. The rejection of claim 16 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks apply to claims 37 and 51.

Regarding claim 17, Tuatini in view of East in further view of Wu fails to teach or suggest the user accessing the client device subsequent to the storing the object and accessing the stored object during the accessing the client device. The Examiner cites paragraph [0043] of Tuatini. However the cited passage makes absolutely no reference to a user accessing a client device subsequent to storing the object or about accessing the stored object during the accessing the client device. Instead, the cited passage describes how Tuatini's application architecture includes an application framework and application and how the application framework receives requests for services from client computers and how action handlers service the requests. The cited passage also describes Tuatini's view handlers that specify the way in which responses are to be presented to client systems. Nothing in the cited passage teaches or suggests a user accessing a client device subsequent to storing the object or about accessing the stored object during the accessing

the client device. Furthermore, neither East nor Wu, either singly or in combination, overcomes this deficiency of Tuatini. Thus, the Examiner's suggested combination of Tuatini, East and Wu fails to teach or suggest the user accessing the client device subsequent to the storing the object and accessing the stored object during the accessing the client device. Therefore, the rejection of claim 17 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks apply to claims 38 and 52.

Applicant also asserts that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

Allowable Subject Matter:

The Examiner indicated that claim 53 would be allowable if rewritten to overcome the rejection under 35 U.S.C. 101 and to include all of the limitations of the base claim and any intervening claims. However, as the rejection of claim 47 has been shown to be unsupported by the prior art, applicants submit that claim 53, which depends from 47, is allowable as currently written.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above referenced application from becoming abandoned, Applicants hereby petition for such an extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-47300/RCK.

Also enclosed herewith are the following items:

- Return Receipt Postcard
- Petition for Extension of Time
- Notice of Change of Address
- Fee Authorization Form authorizing a deposit account debit in the amount of \$ for fees ().
- Replacement Drawing sheets

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: August 9, 2005

In the Drawings:

The enclosed drawing sheets, including FIGs. 18, 23, 30, 33C, 34, 36, 39A, 42B, 45A, 45D and 46 replace the original sheets including the same FIGs. numbers to correct minor margin spacing informalities.